Q1:

```
mov     dword ptr [esp+1Ch], 3
mov     dword ptr [esp+18h], 5
mov     dword ptr [esp+14h], 0
```

This three line declare three variables, their value are 3, 5, 0

```
mov     eax, [esp+1Ch]
imul    eax, [esp+18h]
mov     edx, eax
```

Edx = 3*5

```
mov     eax, [esp+1Ch]
mov     ecx, eax
shr     ecx, 1Fh
add     eax, ecx
sar     eax, 1
```

Eax = 3/2 =1

```
sub     edx, eax
mov     eax, edx
mov     [esp+14h], eax
```

Esp+14h = 3*5 - 3/2 = 14

```
mov     eax, [esp+14h]
mov     [esp+4], eax
mov     dword ptr [esp], offset aD ; "%d"
call    _printf
```

print out calculation result.

Q2

```
.text:0040150E    mov    dword ptr [esp+18h], 0Ch
.text:00401516    mov    dword ptr [esp+1Ch], 0Fh
.text:0040151E    mov    dword ptr [esp+20h], 0DDh
.text:00401526    mov    dword ptr [esp+24h], 3
.text:0040152E    mov    dword ptr [esp+28h], 1B0h
.text:00401536    mov    dword ptr [esp+2Ch], 36h
.text:0040153E    mov    dword ptr [esp+30h], 10h
.text:00401546    mov    dword ptr [esp+34h], 43h
.text:0040154E    mov    dword ptr [esp+3Ch], 0
.text:00401556    mov    dword ptr [esp+38h], 0
```

Initial value for the array, and one variable for res and one variable for index.

```
.text:0040155E                  jmp     short loc_40157F
.text:00401560 ; ---------------------------------------------------
.text:00401560
.text:00401560 loc_401560:                              ; CODE XREF: _main+84↓j
.text:00401560                  mov     eax, [esp+38h]
.text:00401564                  mov     eax, [esp+eax*4+18h]
.text:00401568                  cmp     eax, [esp+3Ch]
.text:0040156C                  jle     short loc_40157A
.text:0040156E                  mov     eax, [esp+38h]
.text:00401572                  mov     eax, [esp+eax*4+18h]
.text:00401576                  mov     [esp+3Ch], eax
.text:0040157A
.text:0040157A loc_40157A:                              ; CODE XREF: _main+6C↑j
.text:0040157A                  add     dword ptr [esp+38h], 1
.text:0040157F
.text:0040157F loc_40157F:                              ; CODE XREF: _main+5E↑j
.text:0040157F                  cmp     dword ptr [esp+38h], 7
.text:00401584                  jle     short loc_401560
```

This is a loop to find the maximum value of the array

```
.text:00401586                  mov     eax, [esp+3Ch]
.text:0040158A                  mov     [esp+4], eax
.text:0040158E                  mov     dword ptr [esp], offset aD ; "%d"
.text:00401595                  call    _printf
```

Print out the maximum value

Q3

```
.text:00401500                  push    ebp
.text:00401501                  mov     ebp, esp
.text:00401503                  and     esp, 0FFFFFFF0h
.text:00401506                  sub     esp, 20h
.text:00401509                  call    ___main
.text:0040150E                  mov     dword ptr [esp+1Ch], 64h
.text:00401516                  jmp     loc_4015D6
.text:0040151B ; ---------------------------------------------------------------------------
```

Assign [esp+1Ch] 100, this value will be used in a loop.

From 1516 to 15d6 I think there is a loop like

```
Int a = 100;
While (a <=999){
        Do some calculation
}
```

```
.text:0040151B loc_40151B:
.text:0040151B                  mov     ecx, [esp+1Ch]
.text:0040151F                  mov     edx, 51EB851Fh
.text:00401524                  mov     eax, ecx
.text:00401526                  imul    edx
.text:00401528                  sar     edx, 5
```

eax /100   get 1th digit of a

```
mov     eax, [esp+18h]
imul    eax, [esp+18h]
imul    eax, [esp+18h]
mov     edx, eax
mov     eax, [esp+14h]
imul    eax, [esp+14h]
imul    eax, [esp+14h]
add     edx, eax
mov     eax, [esp+10h]
imul    eax, [esp+10h]
imul    eax, [esp+10h]
add     eax, edx
cmp     eax, [esp+1Ch]
jnz     short loc_4015D1
```

This part is used to compare the sum of 1st digit ^3 + 2nd digit^3 + 3rd digit^3 == original number (a)

If it is true print out a, otherwise test the next number

From Q4:
Please check the code.

I only submit one c file, this c file contains my code.